架构和开发概要

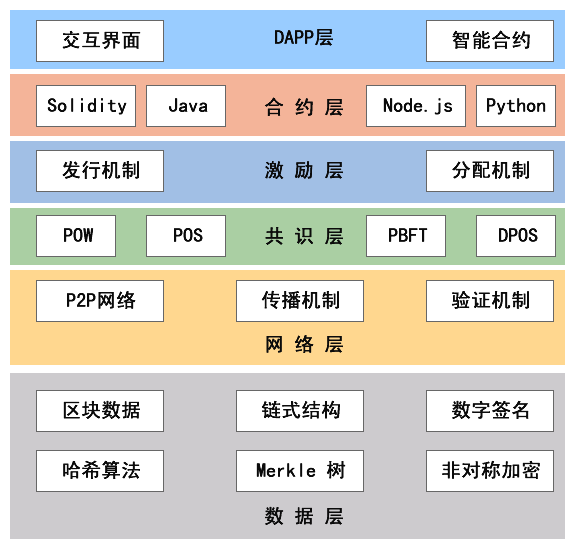
**概述**

区块链是高科技行业，从主机、PC、互联网、移动／社交网络、云计算和大数据之后，出现的新的革命性的变革。

**区块链可以提供什么？**

* + 完整可追溯并且不可更改的交易记录和云账本
  + 去中心化的存储，使得单一节点无法任意修改和垄断数据，减少多方数据共享成本的同时也大大减少灾备成本
  + 通过密码学和其他设计使得在不同交易方中建立信任成为可能
  + 智能合约自动执行事先定义的逻辑，交易方在过程中无法调整更改

1. **分层架构**

****

* **按照技术来分类，从上图可知普通的区块链可以分成：**

**1.数据层**

数据层封装了数据区块的链式结构，区块是区块链的核心单元。区块链由区块互相连接而成。

在区块链中使用了很多加密学算法，包括哈希算法、Merkle树、数字签名等。

* **哈希算法**

哈希算法是一种常见的单向加密算法，它将一串数据加密生成一串二进制，但不能由二进制还原为原来的数据。该算法有一下特点：

相同的输入得到相同的二进制串，不同对输入得到不同的二进制串，即有强对抗性，不同的数据不会产生相同的哈希值，输出的二进制长度是一致的。

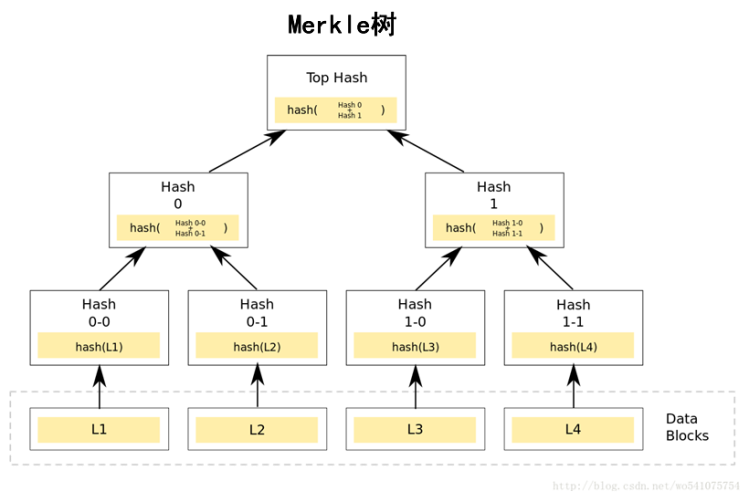
在区块链中很多地方用了哈希算法，比如对区块计算哈希值。

* **Merkle树**

Merkle树，通常也被称作Hash树，顾名思义，就是存储hash值的一棵树。Merkle树是一种数据结构，Merkle树的叶子是数据块(例如，文件或者文件的集合)的hash值。非叶节点是其对应子节点串联字符串的hash。

Merkle树是使区块链发挥作用的基本组成部分。虽然理论上可以在没有Merkle树的情况下制作区块链，但只需创建直接包含每个事务的巨型块头，这样做会带来巨大的可扩展性挑战，可以说无可置疑地使用区块链的能力超出了所有范围，

从长远来看，功能强大的电脑。感谢Merkle树，可以构建在所有计算机和大小笔记本电脑上运行的区块链节点，智能手机，甚至是物联网设备。

****

* **数字签名**

数字签名使用了公钥密码学。公钥密码学是非对称加密技术，其相对于对称加密技术。在对称加密技术中加密使用相同的密钥加解密数据，为了让对方能够解密，需要同时将密文和密钥发给对方。

如果有人在传输过程中截取了密文和密钥，就一样能解密出明文，这就导致了安全性问题。

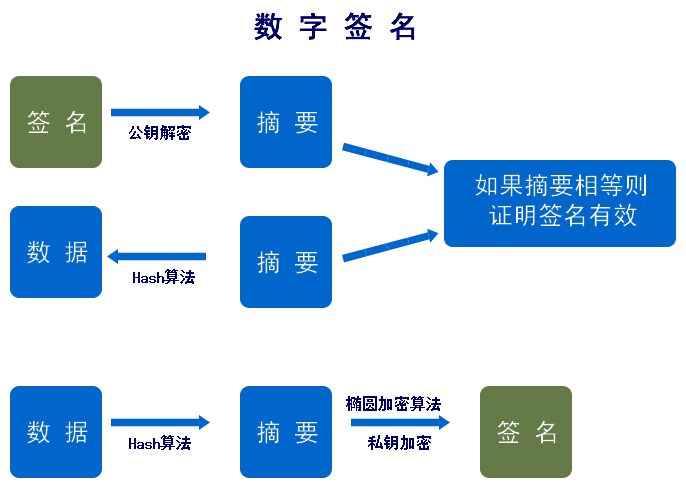
在非对称加密中有公钥和私钥两个，公钥用来加密，私钥用来解密，公钥可以发给任意的人。

在这个过程中只有私钥才可以解密，而加密用的是公钥，故不需要传输私钥，只要保证私钥放在安全的地方，被盗取后其他人还是无法破解密文。

而数字签名就是基于上述非对称加密技术，不同点在于数字签名使用私钥生成一个签名，接收方使用公钥进行校验。比如上面用私钥解密得到明文后，用私钥进行签名进行回复，收到回复后用公钥解密得到的内容与数据相同即可证明签名正确。

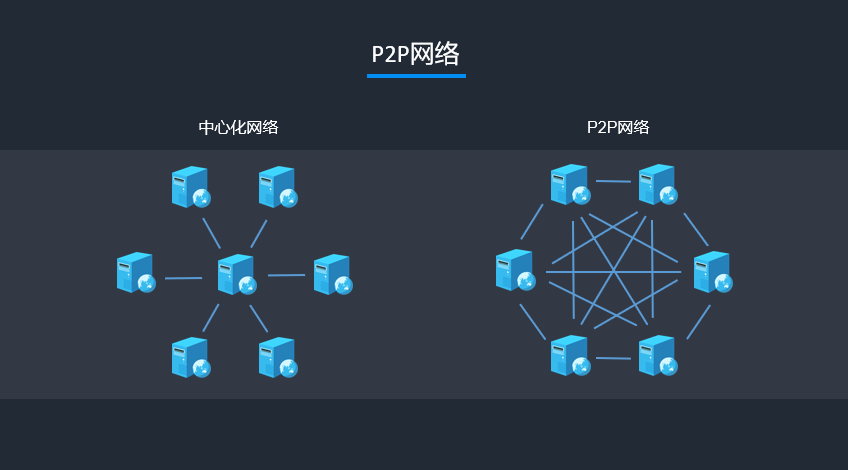
因为公钥和私钥是成对的，唯一对应的，私钥只有对方拥有，所以对方也不能对签名进行抵赖。

在区块链技术中常见的签名算法是椭圆签名算法。其算法用对椭圆曲线上的点进行加法或乘法运算来表达。区块链中私钥是一个随机数，通过椭圆曲线签名算法生成公钥。但反向从公钥计算出私钥几乎是不可能的。椭圆曲线签名算法还具有安全性高和存储空间占用小的特点。

****

**2.网络层**

网络层主要也就是P2P网络。这个不是什么新鲜玩意，最早我们使用的BT(BitTorrent)下载，就是基于P2P网络的，现在很多下载工具都支持。

****

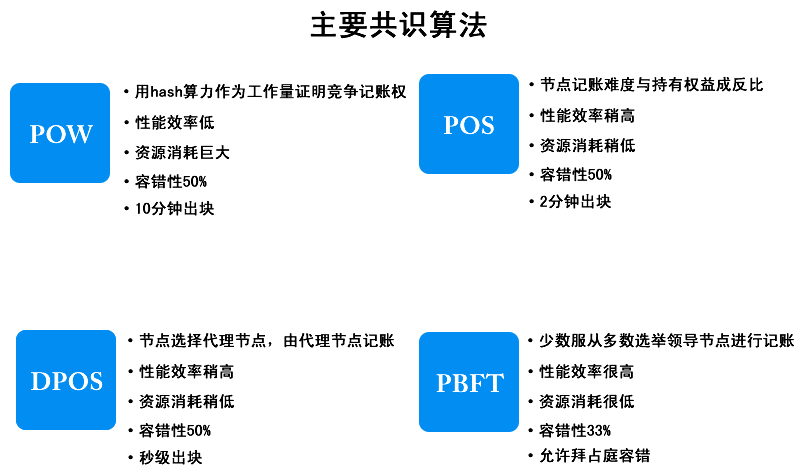
P2P网络并没有一个中心化的节点，所有的节点之间直接互联互通。同时也因为没有中心化的节点，其中任意一点出现问题依旧能够保证整个网络还能通讯，因为你可以连接到其它节点上。

所以 P2P 网络是去中心化的，参与各方地位对等，从而保证网络的健壮性、扩展性、高性能。

**3.共识层**

主要有下面四种共识算法：工作量证明(POW)、权益证明(POS)、委托证明(DPOS)、拜占庭式容错 (PBFT)。

井通SWTC采用改进的PBFT共识。

****

* **工作量证明(POW)**

也就是常见的比特币网络的共识算法，它通过哈希算力作为工作量证明去竞争记账权。POW 的几个缺点：性能比较低；消耗大量的资源，尤其是算力；

出块比较慢，10 分钟左右。但它也优点，抗容错性很好，只有超过 51% 的节点攻击网络，网络才会失败， 容错性 50%。

* **权益证明(POS)**

相对 POW，POS 是用所谓的权益计算法。相对 POW 来说它有几大改进：不再消耗资源；提高性能，2 分钟左右出块；同样容错性也是 50%。

* **委托证明 (DPOS)**

POS 实现 2 分钟左右出块是比较大的提高，但距离业务的要求还有一定距离，所以在此基础上进行优化产生了 DPOS，简单来说 DPOS 不要求每个节点记帐，

而是有代理节点来记帐，这比较像现实生活当中的议会制度，大家选出代表帮忙记帐，这样参与记帐的节点会相对集中，性能也就会提高。所以在 DPOS 情况下基本可以做到秒级出块，容错性也是 50%。

* **拜占庭式容错 (PBFT)**

PBFT 也就是所谓的拜占廷容错，它的性能非常非常快，可以达到秒级出块甚至比秒级还要快，但是它的容错性相对差一些，一般是 33% 左右。



**4.激励层**

主要负责激励的发行制度和分配制度。比如：可以通过挖矿获得token,运行智能合约和发送交易都需要向矿工支付一定的token

**5.合约层**

因为区块链每个区块可编程、可嵌入代码的特性，合约层包含了脚本、算法、以及智能合约，可以简单的理解为是一份自定义的电子合同，之所以称为智能合约，是因为这份合约可以在达到约束条件自动触发执行，不需人工干预，也可以在不满足条件时自动解约。

理论上可以触发执行事先约定好的一切条款。这也是区块链能够解放信用体系最核心的技术之一。

1.将合约以数字化的形式写入区块链中，因区块链的特性，数据将无法删除、修改，只能新增，整个过程透明可跟踪，保证了历史的可追溯性;

2.因行为将被永久记录，可极大程度避免恶意行为对合约正常执行的干扰;

3.去中心化，避免了中心化因素的影响，提高智能合约在成本效率方面的优势;

4.当满足合约内容时，将自动启动智能合约的代码，既避免了手动过程，同时又保障了发行者无法违约;

5.由区块链自带的共识算法构建出一套状态机系统，使得智能合约能够高效地运行。

* **智能合约的执行方式**

以太坊的智能合约执行采用的是同步方式，利用交易触发或者自动触发合约调用，合约在具体执行的时候，区块链的共识机制必须等待合约执行完，返回结果后才能继续操作，从而完成对当前区块的共识。

井通SWTC公链的智能合约在执行上采用了分层、异步调用和快速执行三种机制，其目的是为了保证智能合约执行下共识算法的效率，从而保证井通SWTC公链的商用可行性。

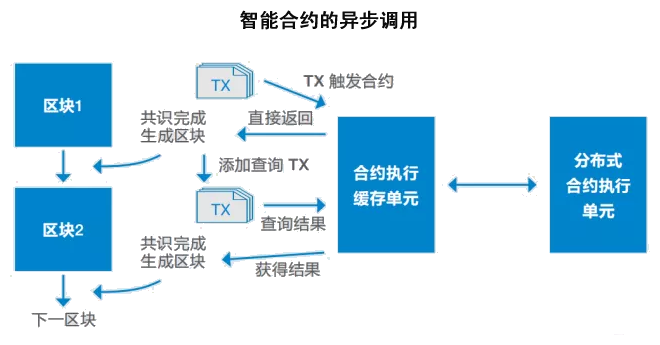
* **井通SWTC公链的区块链系统以异步调用作为后台核心技术，与现有的智能合约执行技术相比具有以下优势：**

1. 隔绝了合约执行和系统共识单元，合约的执行可以在远端，使得合约的执行不再占有系统共识的资源；

2. 合约执行单元和系统共识单元的去耦合，使得合约执行模块和共识模块相对独立，支持热插拔；

3. 在共识验证单元和合约执行单元之间设立合约执行缓存单元，在整个合约执行过程中创造性地采用了异步调用执行方式，使得合约的调用和执行结果在跨区（区块i和区块（i+k））之间分别实现，同时又能保证各个验证节点之间完成共识。该种合约异步调用执行模式提高了合约执行的并发量，共识的过程无需要等待合约的执行结果，大大提高了区块能支持的合约数量；

4. 提高了整个系统的容错能力，一方面系统可以设置合适的超时处理机制来处理合约延时的情况，另一方面，用户可以在合约调用中配置合适的k值，来保证长时间执行的合约得到正确处理。



ETH,MOAC采用Solidity语言部署智能合约。

SWTC公链可采用Node.js,C#等部署合约。

**6.应用层**

以智能合约为核心，添加能够与用户交互的前台界面，形成去中心化的应用（DApp）。

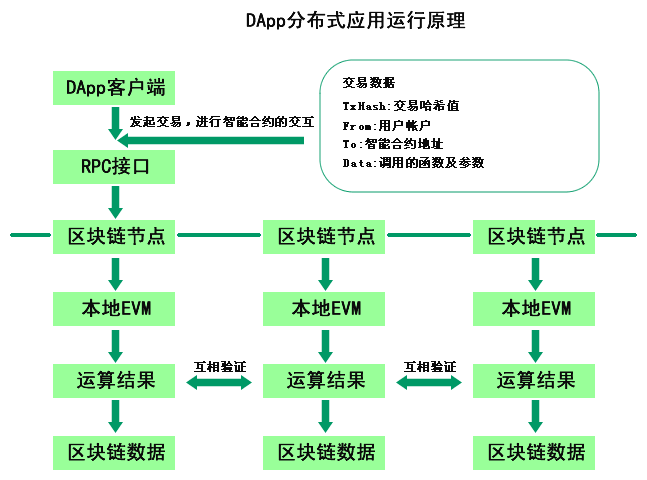
DApp是Decentralized Application的缩写，中文叫分布式应用/去中心化应用，通常来说，不同的DApp会采用不同的底层区块链开发平台和共识机制。

智能合约相当于服务器后台，要实现与用户的友好体验，还需要一些前台页面，通过RPC接口与后台对接，实现网页或者APP访问，部署在服务器上，

拥有完整的智能合约+前台交互界面的组合体称为DApp。

去中心化应用（DApp)的几个特点：

* 应用的数据，分布式存储在区块链的所有节点中；
* 得益于区块链的特性，应用的数据公开透明、不可篡改；
* 应用的逻辑采用智能合约实现，并部署到区块链上；

****



我们以游戏为例，假设APP代表传统游戏，DApp代表区块链游戏。最核心的区别点是：传统游戏的规则和逻辑是厂商制定，

并且运行在中心化的服务器上，厂商随时可以修改升级规则；反之，区块链游戏的规则是通过智能合约，定义并部署在区块链上，规则公开透明，不可篡改；传统游戏的道具和游戏代币，是由厂商控制发行，拥有绝对的掌控权，甚至可以单方面回收玩家游戏内资产；反之，区块链游戏的道具和代币，发行规则完全透明（通过智能合约实现）游戏资产映射到区块链上的TOKEN，资产所有权完全归属玩家，并且可以转账交易，厂商无权控制；

1. **开发概要**

从技术的角度定义区块链 ：区块链是以区块结构存储数据，多方参与、多方维护，通过密码学、P2P网络、共识算法等保证数据可靠的传输、存储、访问的技术体系。

区块链最主要的技术特点：去中心化、不可篡改、可追溯。

**1.开发语言**

 　　区块链是一种编程思想，原则上使用任何一种编程语言都可以实现。比如：C++、Java、python、GO都可以实现开发区块链。

区块链底层系统开发：C++、GO、Python、RUST、JAVA等。

区块链应用系统开发: Solidity、JAVA、Python、RUST、Node.js、PHP、C#等。

GO语言有简洁，快速，安全，并行，开源，支持泛型编程，内存管理，数组安全，编译迅速等特点。有点语言基础，很容易掌握。比特币系统最初使用C++开发，现在也有了GO语言版源码。以太坊官方推出了C++开发版本和GO开发版本。

Rust，其语法类似于C++，它的优势在于它是一种强类型语言（适用于大规模系统开发），并且是具有安全导向的，此外其效率与C++是同一个级别的。越来越多的著名区块链项目已经选择使用Rust作为其开发语言，包括但不限于 Parity, Polkadot, Substrate, Grin, Ethereum classic等等。

Java、Node.js、Python等。这些语言主要是在开发应用程序时，对chaincode(链码/智能合约)进行调用，而且都有与之匹配的SDK。所以语言可以自己选择。因为很多应用是基于Web端的，所以Node.js用得比较多。



**2.环境搭建**

以太坊将智能合约引入区块链，打开了区块链在货币领域以外的应用，从而开启了区块链2.0时代。  
　下面重点讲述以太坊的智能合约开发环境搭建。

* **轻量级开发环境**

如果你只需要开发智能合约（例如开发一个ERC20代币，而不是完整的去中心化应用（DApp），那么使用remix + metamask的轻量级组合就够了。

remix是以太坊官方提供的solidity在线集成开发环境， 你不需要在本地安装任何软件，就可以开发、编译并测试智能合约了。

metamask是一个轻量级钱包，它是chrome浏览器的插件，remix搭配上metamask 以后，你就可以直接在浏览器里部署以太坊智能合约了。

* **DApp快速开发环境**

如果要开发完整的去中心化应用，你就需要部署额外的开发工具了，这主要包括：

Ganache：以太坊仿真器

Truffle：以太坊DApp开发框架

Node.js：后端应用开发平台

Express：node.js的web开发库

这一组合可以让你在一台计算机上模拟以太坊环境，进行DApp的快速开发、部署与测试。 快速的意思是，由于使用了以太坊仿真器，你不需要挖矿、不需要等待交易完成，一切 就像传统的应用开发那么迅速。

* **DApp私有链开发环境**

除了使用以太坊仿真器，你也可以部署一个私有链来开发以太坊应用，这里主要的变化 是采用标准的以太坊节点软件Geth来代替Ganache仿真器。

在私有链开发环境下，你提交到以太坊节点的每一笔交易，需要挖矿才能完成，这意味着你在调试代码时可能需要额外的等待时间，这会拖慢开发的进度。因此，我们建议在可能的情况下，首先使用仿真器先调通你的整个应用，再迁移到私有链环境下

* **DApp公链开发环境**

最终你的开发需要将智能合约部署到公链，例如主链或测试链。这有两种方案：

* + 使用自己的geth全节点
  + 使用公开的托管节点

如果使用自己的geth节点，就和私有链开发环境没有什么区别，你只需要让Geth启动连接到公链而不是你搭建的私有链即可。

但更多的情况是，你并不需要使用自己的全节点，可以利用Infura提供的托管以太坊节点。除非特别需要，我们推荐你首先考虑使用托管节点。

**3.SWTC直连节点开发**

**从上面可以看出从ETH来开发涉及到的知识和技术要求还是挺高的。相比而言，SWTC开发社区提供了相关的API和LIB，来进行快速简单地开发出**满足商业用途的区块链应用。

* <https://github.com/swtcca/swtclib>
* <https://github.com/JCCDex/jcc_rpc>
* 还有正在开发中的：jingtumlib等项目

如果需要了解更多的开发资料可以加入SWTC开发社区QQ群：568285439

**三. 总结**

1. 深入学习比特币、以太坊、SWTC公链等的白皮书。对其技术及原理有全面的了解，并针对其中提到的技术进行深入学习。
2. 搭建比特币、以太坊等的测试网络，对区块系统进行测试。
3. 核心源码学习。这是我们技术实现提升、能够深入理解其技术实现的关键。
4. 学习编写智能合约。多读源码、多练习。
5. 应用开发。主要是通过自己其他语言工具调用智能合约进行技术开发。目前SWTC公链已提供的基于Node.js、JAVA、PHP、Python等语言的SDK，这些SDK对区块链中常用操作进行了封装，已大大简化了开发工作。